

ABSOLUTE POSE REGRESSION FOR LOCALIZATION USING MULTIPLE IMAGE STREAMS

A PREPRINT

 **Vaibhav Arora***
Informatique
Université Paris Saclay
Orsay, FR 91400
vaibhav.arora@u-psud.fr

October 11, 2021

ABSTRACT

Localization of a user or any autonomous system can be performed using deep learning with typical sensor data like RSS from WAPs, inertial data (accelerometer, gyroscope, magnetometer), etc. Methods pertaining to such data is susceptible to large errors under certain conditions due to the nature of the data (noise, lack of infrastructure, differences in sensors, etc). On the other hand, many end-to-end methods have shown promising results on regressing the 6-DOF pose from RGB images directly. These methods have their own pitfalls too (for example, textureless or pattern-repetitive scene). In this work, we propose the usage of image streams from multi-angled cameras for absolute pose regression (APR). We focus on improving the baseline of APR using single image streams by suitable methods fusing multiple-image streams. We evaluate on the Hyundai dataset and show that input fusion outperforms fusion at some intermediate embedding, and the naive APR approaches based using single image streams. We also show that usage of a loss function capturing the relative motion prior and data augmentation simulating different directions further improves the performance gain for multi-image stream methods.

Keywords APR · Fusion · Transformers

1 Introduction

Localization of a user or an autonomous system plays a central role in many applications involving navigation specifically in indoor settings where the mature global-positioning systems (GPS) are obstructed. This task becomes more challenging when the cost of the sensors or infrastructure involved is to be kept low, which in turn prohibits use of for example, lidars. Typical approaches then include use of received-signal strength (RSS) from devices such as WiFi, using magnetic sensors, and more recently camera sensors which have shown promising results by determining the absolute pose, i.e. the position and orientation of the camera given an image. This last method is known as camera pose estimation and the current state-of-the-art are based on structure based methods.

In structure based methods, the camera pose is estimated using a Perspective-n-point (PnP) solver inside a RANSAC loop. The 2D-3D correspondences are computed by matching local image features (cite sarlin). The downside of these approaches is that these methods are memory intensive, adaption to newer scenes is difficult and inference is much slower. Absolute pose regression (APR) methods employ single models mapping pixels directly to pose typically employing convolutional networks (CNNs). This makes them lightweight and inference is an order of magnitude faster but these methods lag behind the structure based methods in accuracy of the estimated pose. APR thus remains an

*Use footnote for providing further information about author (webpage, alternative address)—*not* for acknowledging funding agencies.

active research area with various improvements being proposed mainly in terms of the architecture and the loss functions.

A common characteristic of all the previous approaches has been that the pose is estimated using a single image. This is in part due to the fact that all previously available datasets for camera localization are based on images from single camera for a given scene cite 7scenes, cambridge, oxford. In contrast to this, the Hyundai departmental store dataset cite NLE for visual localization was captured using a dedicated mapping platform with ten cameras (six Basler cameras and four smartphones). Thus for any co-ordinate, four to six images or even ten can be used for localization. This richer set of input space potentially gives better features to any function transforming pixels to pose. An increase in performance is should therefore be expected.

In this work, we empirically explore fusion of multiple images for APR. Given the success of the previous methods on this task cite posenet mapnet detr, we expand them to incorporate multi-angled images. Different fusion architectures are evaluated on the Hyundai dataset and compared to the baseline showing improvement in performance. Further gains are achieved by employing the relative pose loss cite mapnet to these fusion architectures and a specific data augmentation unique to the setup of multi-angled cameras.

In summary, the main contributions of this work are as follows:

- Empirical exploration of fusion architecture to use multiple images from different cameras (facing different directions) for APR giving improvement over the baseline.
- Use of relative loss
- Data augmentation to simulate different heading of the mapping platform

2 Related Work

2.0.1 Absolute pose regression

There are multiple works that use deep neural networks for image based localization. PoseNet [7] regresses the 6-DOF camera pose from a single RGB image in an end-to-end manner without the need of additional engineering or graph based methods. It was found robust to difficult lightning, motion blur and different camera-intrinsic parameters where the traditional point based SIFT methods fail. The authors modify GoogleNet pre-trained on ImageNet dataset and fine tune it. The 3 softmax classifiers are replaced with regressors. Each final fully connected layer (2048) is modified to output a pose vector of 7-dimensions representing position (x-y coordinates) and orientation (a quaternion). They use an affine combination of Euclidean loss for the position and orientation error scaled by some factor chosen to keep the expected value of the position and orientation error to be approximately equal. The authors claimed that context and FoV of the image is more important than the resolution for relocalization. Training PoseNet required expensive tuning of the hyperparameter scale factor. The same authors extend PoseNet by learning the weight between camera translation and rotation loss and incorporating the reprojection loss [6].

But PoseNet is significantly less accurate than state-of-the-art SIFT based methods. Authors of the work [13] propose a novel CNN+LSTM architecture for camera pose regression which performs better than PoseNet and performs well in hard conditions (textureless surfaces) where SIFT-based methods completely fail. Similar to PoseNet, they make use of pre-trained GoogleNet and removal of softmax classifiers with regression layer. But instead of 7-D pose regression after the fully-connected layer, they make use of 4 LSTM units after the FC layer. LSTM units on the CNN output play the role of structured dimensionality reduction on the feature vector. The authors claim that this avoids overfitting (other dimensionality reduction methods can be used too but use of LSTMs was found to perform better) and results in improvement of the localization performance.

Prior DNNs for camera localization are trained using single images labelled with absolute camera pose. The authors of MapNet [1] show that geometric constraints between pairs of observations can be included as an additional loss term to the original PoseNet loss term (camera motion-geometry aware learning) which significantly improves camera localization performance. The authors make modifications to PoseNet. First, they use ResNet-34 and modify it by introducing a global average pooling layer after the last convolution layer, followed by a FC layer with 2048 neurons, a ReLU and dropout. This is followed by a final FC layer that outputs a 6-DoF camera pose. The main difference, however, is that MapNet minimizes both the loss of the per-image absolute pose and the loss of the relative pose between image pairs.

State-of-the-art algorithms follow a 3D structure-based approach. Classic SIFT (or their recent improvements) approaches outperform all published CNN based pose regression methods to date [13, 10]. Although structure based methods can completely fail in some challenging environments [7, 13], absolute pose regression does not perform well at all for the particular dataset of interest for this work (henceforth referred to as the Hyundai dataset) and structure based methods significantly outperform coordinate point regression and absolute pose regression approaches [8]. However, the latter use SfM (Structure-from-Motion) models and therefore have a requirement of constructing and maintaining 3D maps (2D-3D matches from descriptor matching are used to estimate the camera pose by applying PnP solvers inside a RANSAC loop). Whereas the task in hand is constrained to use only sensors available in a smartphone. More importantly, the shortcomings of deep learning based absolute pose regression can be met by other sensors via fusion which is the main scope of this work.

In fact no current pose regression approach consistently outperforms structure-based methods. The authors of [10] develop a theoretical camera pose regression model (APR) and show that APR methods learn a set of base poses such that poses of all training images can be expressed as a linear combination of these base poses. The network learns to sum these base poses up to an absolute pose by scaling them appropriately via the coefficients in the embedding (the output of the second-to-last layer). They are bound to interpolate poses in the training data for the test data and thus not guaranteed to generalize. The authors validate their claim through clever experiments by collecting training data in a line and test data in a line parallel to the train data. They then move on to show that all the predictions of the poses on the test data lie in the span of the training data line.

2.0.2 Relative pose regression

There is the approach of relative pose regression which combines the methods of camera pose regression with an image retrieval (IR) scheme. Authors of [15] propose a 3D model-free localization pipeline based on essential matrices. Another noteworthy work is [9] where the authors introduce an end-to-end trainable approach PixLoc which leverages 3D maps with classic image alignment where a residual between the CNN based extracted features projected onto the 3D map is minimized between the query image and an image from the database (using a image retrieval method). This method yields results on par with structure based methods and generalizes across different datasets and is expected to perform well on the Hyundai dataset too. The authors argue that the deep network does not need to learn geometric principles (unlike in the case of APR methods) and rather focus on getting robust features. Although giving superior results, this split of the task from the deep network is not useful to retrieve an embedding which can be used for fusion with data with other modalities. Moreover a need for image database is not desirable to transfer to new scenes. We therefore stick to APR based methods in this work.

2.0.3 Transformers based APR

All the methods discussed so far employ a convolutional backbone which is typical for tasks involving images. Recent works have shown the success of Transformers [12] on various vision tasks [4], [2]. Following the trend, authors of [11] claim that the current best performing models use attention mechanism and show that their method of using a detection transformer (DETR) [2] based backbone allows to learn multi-scene absolute pose regression with best results on the 7Scenes dataset [5].

Unique to our dataset is the availability of multiple images from cameras facing different directions for each co-ordinate. In this work, focus on improving the baseline of APR using single image streams by proposing a suitable method for fusing multiple-image streams surpassing the baseline performance. Our method builds on top of the DETR backbone [2], [11].

2.1 Problem statement

Given a dataset of images $\mathbf{X} \in \mathbb{R}^{N \times C \times H \times W}$, the typical camera pose estimation task is to use a mapping function $f(X)$ to regress the pose $\mathbf{Y} \in \mathbb{R}^{N \times 7}$ in a supervised manner minimizing an empirical loss $\mathbb{L}(f(x_i), y_i)$ with $i \in 1, 2, \dots, N$. Here N is the number of training examples and each output vector consists of the co-ordinate $\langle x, y, z \rangle$ and the quaternion $\langle w, p, q, r \rangle$.

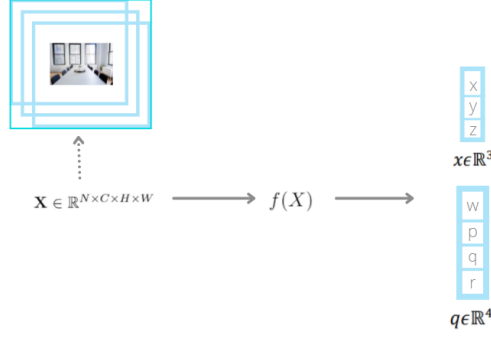


Figure 1: Camera pose estimation

3 Absolute Camera Pose Regression using Multiple Image Streams

$f(X)$ typically takes the form of a CNN (cite posenet mapnet pixloc) but more recently transformers have been included (cite detr). The usual loss function is the \mathbb{L}_1 distance between the ground-truth and estimated pose and more recently the inclusion of relative pose error over a tuple of image sequences has shown improvement in results (cite mapnet).

$$\mathbb{L} = \sum_{i=1}^N h(p_i, p_i^*) + \sum_{i,j=1, i \neq j}^N h(v_{ij}, v_{ij}^*)$$

But all these existing approaches make use of a single image to estimate the pose. In our multi-angled camera setup, each training sample consists of a tuple of images such that $\mathbf{X} \in \mathbb{R}^{N \times T \times C \times H \times W}$ where T is the number of available cameras. Each image corresponding to a given camera facing a particular direction in the tuple thus provides complementary information. The goal is then to fuse these multiple image streams for regressing pose. The ability to represent data from different modes in a meaningful way is thus crucial (cite morency). This is similar to the well established multimodal fusion problems, only the data is homogeneous. Since in tasks relying on presence of data both during training and inference use joint representations, we follow this approach. Joint representations combine data from each mode or (image stream in our case) to the same representation space (cite morency). For the i^{th} tuple of images, this joint representation can be expressed as:

$$\mathbf{r}_i = f(g(x_{c_1}) \oplus g(x_{c_2}) \oplus \dots \oplus g(x_{c_T}))$$

where $g(\cdot)$ can either be identity (for fusion at input) or some function and $f(\cdot)$ then would be another function operating on the joint embedding. The usual approach is to split existing backbones at the point of fusion, then $g(\cdot)$ representing the first half of the network and $f(\cdot)$ representing the second half.

3.1 Proposed approach

Dataset. Recently released publicly (cite NLE) we use the Hyundai departmental store dataset 4F (4th floor) using images from only the four galaxy smartphones for the evaluation of the various models explored. Later we also evaluate the best model on the 1F (1st floor) dataset and the images from the six Basler cameras (see (cite NLE) for details).

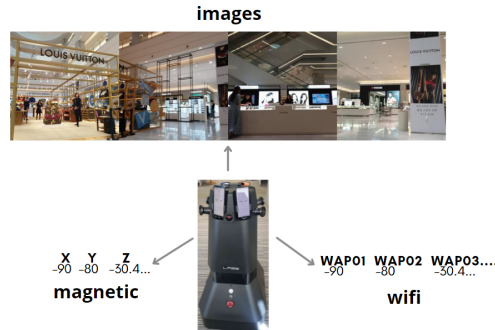


Figure 2: In-house mapping platform

Architecture backbone. Any fusion network relies on a backbone network where each successive layer of the backbone network extracts some intermediate representation of the data which is then fused. Backbones involving CNNs include PoseNet (cite posenet), PoseLSTM and MapNet (cite mapnet). Of these MapNet is more recent with its inclusion of relative loss and is known to perform better. We thus stick to MapNet. Backbones including transformers like TransPoseNet are based on detection transformers (DETR) which have also shown to be best performing on the 7-Scenes dataset (cite shovit). We thus explore different joint fusion networks based on MapNet and DETR.

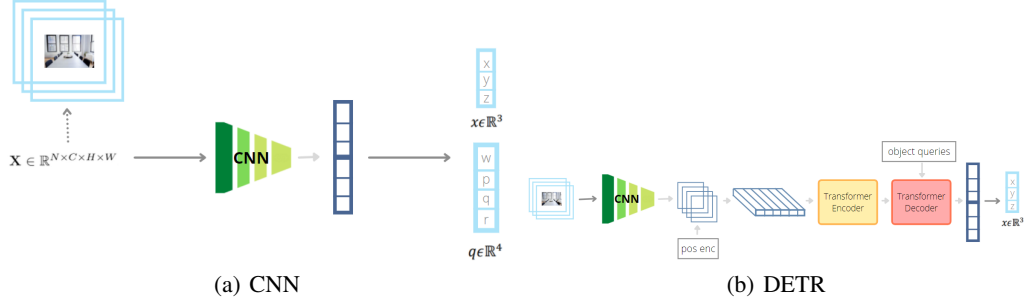


Figure 3: Typical backbone architectures for APR

Fusion. To get the joint representation, typical approach follows fusion by concatenation, in which we concatenate the output features or some intermediate embeddings from the individual networks for each mode (or image stream in our case). Another approach is to perform fusion of the input data directly before feeding it to the network. We explore all these approaches of late-fusion, mid-fusion and input-fusion suitably for both MapNet and DETR based backbones where the position of fusion being dependent on the model footprint on the memory to avoid model parallelism.

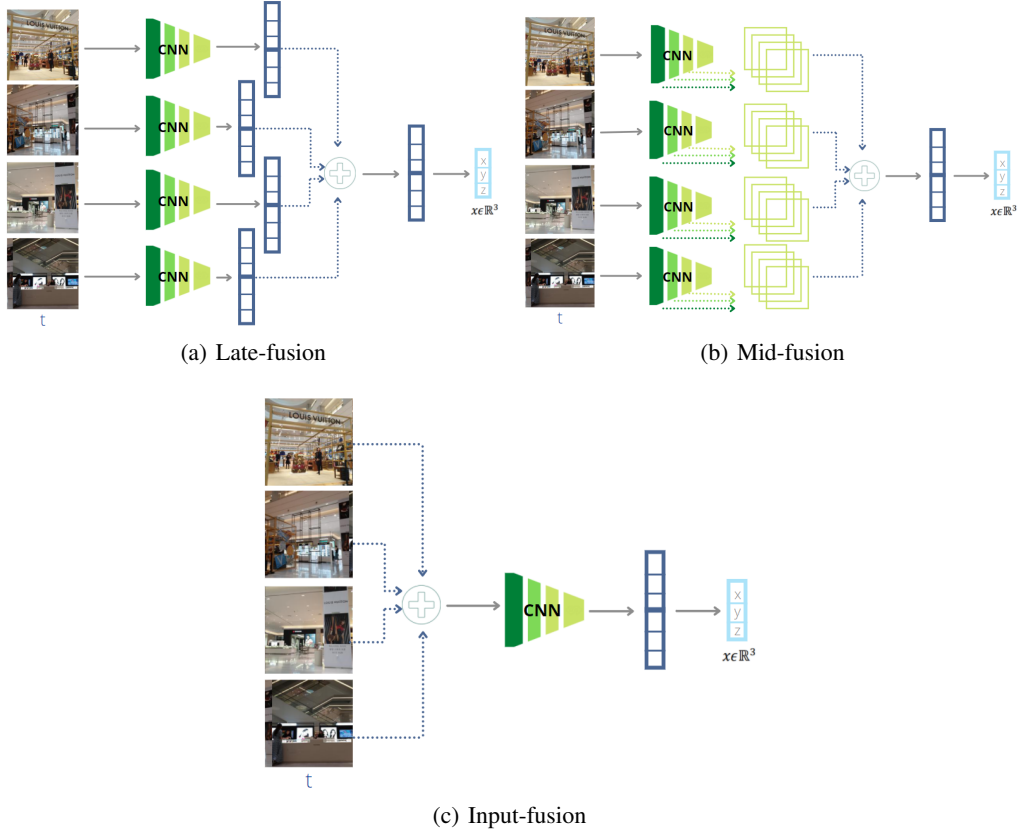


Figure 4: Different fusion architectures with a CNN based backbone network

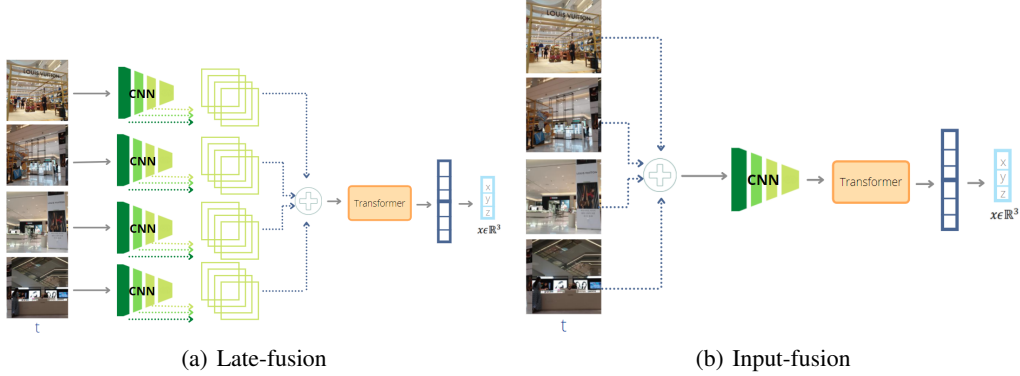


Figure 5: Different fusion architectures with a DETR based backbone network

4 Experiments and analysis

4.1 Baseline with mix-stream input

In line with our goal, we want to find a suitable joint representation of multiple image streams providing complementary information for the same output pose. To this extent, we first explore the performance of baseline models on our dataset without fusion of the data or its embedding, therefore the input dataset includes all the images from all the cameras which we henceforth refer to as 'mix-stream'. Note that images from all cameras are used and not one since at test time, the image can be in any direction and therefore using single image streams is sub-optimal. As we can see from 1, MapNet is the best performing model with CNN in backbone. This is expected since MapNet essentially builds on top of PoseNet with the inclusion of a relative loss in addition to the absolute pose loss. Overall the DETR based model TransPoseNet performs the best. TransPoseNet employs separate positional and orientational encoders and decoders for aggregation of the feature maps from a CNN and able to attend to localization-informative image content: corners and blob-like cues are positional informative and elongated corners are emphasized by orientational encoders [11]. Note that regressing position and orientation separately with PoseNet like approach has already been found to be sub-optimal and therefore we skip these experiments [7]. Vision Transformers (ViT) [3] show competitive results as TransPoseNet.

Table 1: Baseline model performance on mix-streams

Model	Mean [m]	Median [m]
PoseNet	5.39	3.52
PoseLSTM	5.32	3.21
MapNet	4.49	3.03
TransPoseNet	2.33	1.175
ViT	2.493	0.936

4.2 Multi-stream fusion

Considering the best baseline models as backbone for the fusion architecture, we explore joint fusion by concatenation at different positions constrained by the GPU memory requirements. From 2, we can see that for the CNN based architectures, fusion of the final embeddings gives the best performance and on the other hand, fusion at early CNN blocks gives the worst performance. This is likely due to final embedding representing being closest of the output space that the model learnt and the individual models fail to extract any relevant structure in the early CNN blocks. With the DETR based backbone, fusion at input gives the best performance overall. The gap in mean between input fusion and fusion post-CNN can be explained from the fact that the test dataset was likely taken in another direction. Therefore individual models trained with one set of features see different features on test time. This led to the data augmentation as explained in the section of Track Augmentation. This won't be a problem when the fusion occurs at input since a single model would be trained with the input features from all the cameras and would learn to extract the relevant ones. A proven technique of Gradient Blending [14] for heterogeneous multimodal data was also given a shot, but performs worse. This is likely due to the homogeneity of the data resulting in similar overfitting rates and thus similar blending

weights which won't contribute to the 'blending' of the individual losses from the individual modalities, resulting in one big over-parameterized model.

Table 2: Model performance with fusion at various positions

Backbone	Fusion position	Mean [m]	Median [m]
MapNet	fc-2	3.92	3.15
	fc-1	4.07	3.31
	fc-0	2.60	1.87
	block-5	3.35	2.71
	block-4	3.26	2.48
	block-3	3.99	3.12
	block-2	6.05	4.06
TransPoseNet	input	1.68	1.29
	CNN	4.94	2.02
ViT	input	2.98	1.82
Gradient-blending	fc-0	5.10	3.63

4.3 Relative Loss

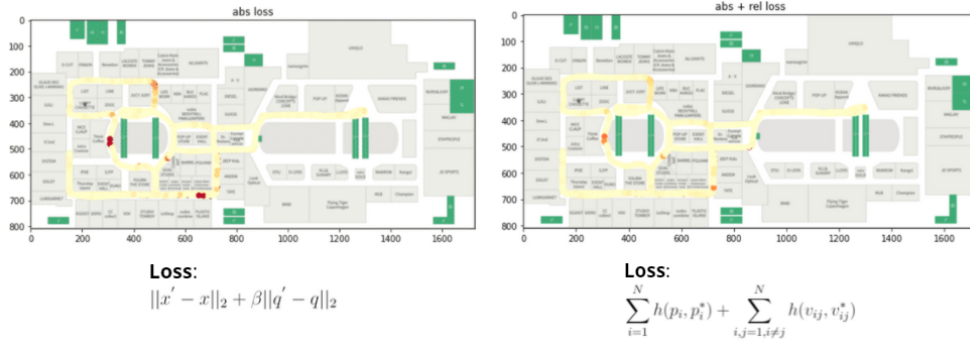


Figure 6: Impact of using relative loss

Apart from introducing inductive priors via changes in model architecture, it is possible to also include priors via the loss function. For any physical system, the pose at time t and time $(t + 1)$ would be relatively similar. Inclusion of a relative pose loss term in addition to the absolute loss can thus provide a better training signal for the model. First introduced in [1], we use the relative loss with our best model and observe that it consistently improves the performance (see Table 3).

Table 3: Impact of using absolute and relative loss

Backbone	Fusion position	Loss	Mean [m]	Median [m]
TransPoseNet	-	absolute	2.33	1.17
	input 4-streams	absolute	1.68	1.29
	-	relative	1.98	0.86
	input 4-streams	relative	1.306	0.985

4.4 Data augmentation: simulating new tracks

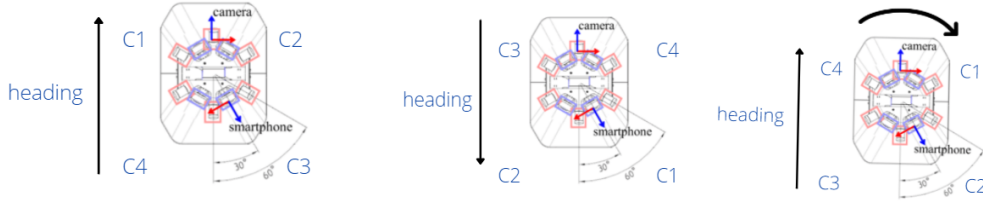


Figure 7: Track augmentation by switching camera positions

It can be shown that end-to-end APR based camera pose estimation methods learn base poses from the training data and then interpolate from the base poses during inference time, based on the closest features between the test and train samples, in an image retrieval approach [10]. Therefore the test distribution must be a subset of the train distribution. In our setting, the coverage of the train distribution can be improved via simulating new tracks, by switching the positions of the cameras during training. This simulates different heading directions of the robotic platform. Much more than architectural changes, we observe (see Table 4) that this simple data augmentation improves the performance significantly.

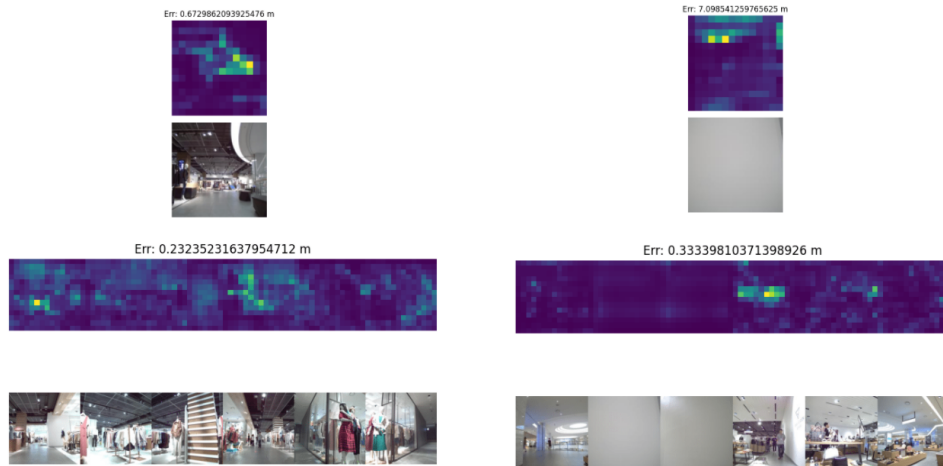


Figure 8: Attention maps for single images and multi-images fused at input

From the decoder attention maps in figure 8, we can see that when model has few features to pick upon, the error in pose is high. This is because images with few unique features in different parts of the scene (like a plain white wall, for example) would all be mapped to a similar output space, resulting in a long tailed distribution of errors. With the use of multiple images for the same co-ordinate, the model has more context to pick up from and thus always has unique features to map to unique spaces. Simulating new tracks increases the set of unique features that the model can see and thus leads a way to an indirect form of generalization.

Table 4: Impact of track augmentation

Backbone	Fusion position	Track augmentation	Mean [m]	Median [m]
TransPoseNet	-	no	2.33	1.17
	-	yes	2.42	1.14
	CNN	no	4.94	2.02
	CNN	yes	2.10	1.53
	input 4-streams	no	1.68	1.29
	input 4-streams	yes	1.272	0.989

4.5 Applicability to other datasets

To ensure our methods don't overfit to the one dataset used, we try the methods on different datasets. From Table. 5 and 6, we can see that using these different strategies of multiple-images for a specific co-ordinate, use of relative loss term and simulating new tracks in conjunction improves the baseline performance significantly.

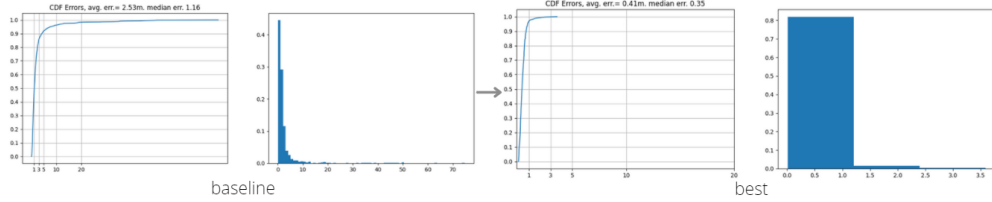


Figure 9: Overall improvement in baseline performance

Table 5: Baseline and best model on different datasets

Model	galaxy-4F		Basler-4F		galaxy-1F	
	Mean [m]	Median [m]	Mean [m]	Median [m]	Mean [m]	Median [m]
baseline	2.33	1.175	1.946	0.927	4.839	2.790
best	1.087	0.813	0.715	0.594	2.966	2.073

Table 6: Best model on 4F-dataset with Basler images using different strategies

Model	Fusion	Relative Loss	Track Augmentation	Mean [m]	Median [m]
TransPoseNet	-	-	-	1.946	0.927
	Yes	Yes	-	0.715	0.594
	Yes	Yes	Yes	0.382	0.333

5 Conclusion

In this work, we empirically explored different backbone architectures and implemented different fusion strategies for pose regression with multiple images. A well known problem of APR based camera pose estimation is that close to image retrieval, such models learn base poses and then interpolate for features of test samples that are similar to features of the training samples. This results in a long tailed error distribution since different images in different parts of the scene would be mapped to a similar output space. We observed that transformer based backbones with input fusion perform the best due to their global feature aggregation from self-attention, important when we have multiple images for the same co-ordinate and we want to benefit from the complimentary data from different images in the form of more unique sets of features to be mapped to unique output spaces, decreasing the typical long tailed error distribution of APR based camera pose estimation. Moreover, specific data augmentation of switching camera positions during training simulates new tracks giving an overall performance boost, along with the additional usage of a relative pose loss term. These different strategies in conjunction improve the baseline results of typical APR models trained using single images.

References

- [1] Samarth Brahmabhatt et al. “Geometry-Aware Learning of Maps for Camera Localization”. In: *arXiv:1712.03342 [cs]* (Apr. 2018). arXiv: 1712.03342. URL: <http://arxiv.org/abs/1712.03342> (visited on 05/20/2021).
- [2] Nicolas Carion et al. “End-to-End Object Detection with Transformers”. In: (2020). arXiv: 2005.12872 [cs.CV].
- [3] Alexey Dosovitskiy et al. “An Image is Worth 16x16 Words: Transformers for Image Recognition at Scale”. In: *CoRR* abs/2010.11929 (2020). arXiv: 2010.11929. URL: <https://arxiv.org/abs/2010.11929>.
- [4] Alexey Dosovitskiy et al. “An Image is Worth 16x16 Words: Transformers for Image Recognition at Scale”. In: (2021). arXiv: 2010.11929 [cs.CV].
- [5] Ben Glocker et al. “Real-Time RGB-D Camera Relocalization”. In: *International Symposium on Mixed and Augmented Reality (ISMAR)*. Edition: International Symposium on Mixed and Augmented Reality (ISMAR). IEEE, Oct. 2013. URL: <https://www.microsoft.com/en-us/research/publication/real-time-rgb-d-camera-relocalization/>.
- [6] Alex Kendall and Roberto Cipolla. “Geometric Loss Functions for Camera Pose Regression with Deep Learning”. In: *arXiv:1704.00390 [cs]* (May 2017). URL: <http://arxiv.org/abs/1704.00390> (visited on 05/04/2021).
- [7] Alex Kendall, Matthew Grimes, and Roberto Cipolla. “PoseNet: A Convolutional Network for Real-Time 6-DOF Camera Relocalization”. In: *arXiv:1505.07427 [cs]* (Feb. 2016). URL: <http://arxiv.org/abs/1505.07427> (visited on 05/04/2021).
- [8] Donghwan Lee et al. “Large-scale Localization Datasets in Crowded Indoor Spaces”. In: *arXiv:2105.08941 [cs]* (May 2021). URL: <http://arxiv.org/abs/2105.08941> (visited on 05/31/2021).
- [9] Paul-Edouard Sarlin et al. “Back to the Future: Learning Robust Camera Localization from Pixels to Pose”. In: *CVPR*. 2021. URL: <https://arxiv.org/abs/2103.09213>.
- [10] Torsten Sattler et al. “Understanding the Limitations of CNN-Based Absolute Camera Pose Regression”. In: 2019, pp. 3302–3312. URL: https://openaccess.thecvf.com/content_CVPR_2019/html/Sattler_Understanding_the_Limitations_of_CNN-Based_Absolute_Camera_Pose_Regression_CVPR_2019_paper.html (visited on 05/28/2021).
- [11] Yoli Shavit, Ron Ferens, and Yosi Keller. “Learning Multi-Scene Absolute Pose Regression with Transformers”. In: *arXiv:2103.11468 [cs]* (July 2021). arXiv: 2103.11468. URL: <http://arxiv.org/abs/2103.11468> (visited on 08/10/2021).
- [12] Ashish Vaswani et al. “Attention Is All You Need”. In: *arXiv:1706.03762 [cs]* (Dec. 2017). arXiv: 1706.03762. URL: <http://arxiv.org/abs/1706.03762> (visited on 07/16/2021).
- [13] Florian Walch et al. “Image-based localization using LSTMs for structured feature correlation”. In: *arXiv:1611.07890 [cs]* (Aug. 2017). URL: <http://arxiv.org/abs/1611.07890> (visited on 05/15/2021).
- [14] Weiyao Wang, Du Tran, and Matt Feiszli. “What Makes Training Multi-Modal Networks Hard?” In: *CoRR* abs/1905.12681 (2019). arXiv: 1905.12681. URL: <http://arxiv.org/abs/1905.12681>.
- [15] Qunjie Zhou et al. “To Learn or Not to Learn: Visual Localization from Essential Matrices”. In: *arXiv:1908.01293 [cs]* (Mar. 2020). arXiv: 1908.01293. URL: <http://arxiv.org/abs/1908.01293> (visited on 07/15/2021).